# Unique Decomposition of Processes

Theoretical Computer Science 107 (1993) 357–363
Elsevier



Note

# Unique decomposition of processes*

Robin Milner and Faron Moller

*Department of Computer Science, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK*

Communicated by G. Rozenberg
Received October 1991

*Abstract*

Milner, R. and F. Moller, Unique decomposition of processes, Theoretical Computer Science 107 (1993) 357–363.

In this paper, we examine questions about the prime decomposability of processes, where we define a process to be prime whenever it cannot be decomposed into nontrivial components.

We show that any finite process can be uniquely decomposed into prime processes with respect to bisimulation equivalence, and demonstrate counterexamples to such a result for both failures (testing) equivalence and trace equivalence.

Although we show that prime decompositions cannot exist for arbitrary infinite processes, we motivate but leave as open a conjecture on the unique decomposability of a wide subclass of infinite behaviours.

## 1. Introduction

Let $\|$ be a binary operator for putting two processes together in parallel, which is commutative and associative and has a unit. Then there is an obvious definition of *prime* process, and an obvious question whether, for a given process $P$, there is a unique multiset $\{A_1, \ldots, A_n\}$ of primes for which

$$P = A_1 \| A_2 \| \cdots \| A_n.$$

*Correspondence to*: F. Moller, Department of Computer Science, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK. Email: fm@lfcs. ed.ac.uk.

* Appeared in the Bulletin of the European Association for Theoretical Computer Science, June 1990.

But there seems to be very little known about such questions.

There are several degrees of freedom: What class of processes are we considering? Precisely which operator $\|$ are we considering? What notion of equality or congruence does $=$ stand for?

In this note we answer the question for a class of finite processes, with a natural parallel operator; the answer is positive for one congruence, but negative for two others (thanks to Rob van Glabbeek and Joram Hirshfeld). We also conjecture a positive answer for a class of *regular* (i.e. finite-state) processes; as soon as we leave the domain of finite processes, the question seems to get much harder and highly intriguing. We hope others will find it so, and come up with some answers where we have failed so far.

The question does not seem an idle one. Surely, with the appropriate parallel operator, the answer is relevant to the way in which many processors can be usefully deployed upon a problem – at least, under static allocation of processors to subproblems.

## 2. Finite processes

We consider here a language $\mathscr{P}$ of *process terms*, namely the set of terms over the signature $\Sigma = \{0, ., +, \|\}$; 0 represents the nil process, . represents prefixing of actions taken from some set $Act$, $+$ represents nondeterministic choice, and $\|$ represents full merge. We adopt the usual operational semantics of this simple language, namely the least transition relation $\rightarrow \subseteq \mathscr{P} \times Act \times \mathscr{P}$ (writing $P \xrightarrow{a} Q$ for $(P, a, Q) \in \rightarrow$) such that $a.P \xrightarrow{a} P$, and such that $P \xrightarrow{a} P'$ implies each of $P+Q \xrightarrow{a} P'$, $Q+P \xrightarrow{a} P'$, $P \| Q \xrightarrow{a} P' \| Q$, and $Q \| P \xrightarrow{a} Q \| P'$. The semantic congruence which we consider is strong bisimilarity $\sim$ [4]; this is the largest binary relation on terms such that $P \sim Q$ if and only if, for all $a \in Act$,

- $P \xrightarrow{a} P'$ implies $Q \xrightarrow{a} Q'$ for some $Q'$ such that $P' \sim Q'$; and
- $Q \xrightarrow{a} Q'$ implies $P \xrightarrow{a} P'$ for some $P'$ such that $P' \sim Q'$.

We rely on the well-developed theory for this language and congruence, which tells us that the congruence is completely characterized as isomorphism between derivation trees, finite unordered trees whose arcs are labelled by elements of the action set $Act$, in which no two identically labelled arcs lead from the same node to two isomorphic subtrees. Another characterization is that $\mathscr{P}/\sim$ is the initial $\Sigma$-algebra satisfying the laws of a commutative monoid with absorption – $P+Q=Q+P$, $P+(Q+R)=(P+Q)+R$, $P+0=P$ and $P+P=P$ – and an expansion law relating $\|$ to the other operators.

The proof that follows will proceed by induction on the *size* $|\cdot|$ of a term, given by the *depth* of its derivation tree:

$$|0| = 0, \qquad |P+Q| = \max(|P|, |Q|),$$

$$|a.P| = 1 + |P|, \qquad |P \| Q| = |P| + |Q|.$$

Equality throughout this note will represent semantic equality (strong bisimilarity). Thus, $P=Q$ will mean $P\sim Q$; if we need to consider the syntactic identity of terms, we write $P\equiv Q$.

The important properties which we shall use are as follows, and are immediate consequences of the definitions:

- $P=Q$ implies $|P|=|Q|$;
- $P\neq 0$ implies $|P\parallel Q|>|Q|$;
- if $P=Q$ and $P\xrightarrow{a}P'$ then $Q\xrightarrow{a}Q'$ for some $Q'=P'$;
- $P\xrightarrow{a}P'$ implies $|P|>|P'|$.

**Definition 2.1.** A term $P$ is *irreducible* if whenever $P=Q\parallel R$, we have that either $Q=0$ or $R=0$.

**Definition 2.2.** A term $P$ is *prime* iff $P$ is irreducible and $P\neq 0$.

We shall now prove that unique decomposition into primes exists, up to $\sim$. The original proof of this result, by Milner, proceeded directly by induction on size. The case analysis was rather detailed; so, we prefer to give here Moller's shorter proof, which proceeds via a cancellation lemma. Both proofs were first reported in [5], where the result is also extended to allow synchronized communication between parallel processes.

**Lemma 2.3** (Cancellation). *For $P$, $Q$ and $R\in\mathscr{P}$,*

$$P\parallel R=Q\parallel R \text{ implies } P=Q.$$

**Proof.** We actually prove the following two results by simultaneous induction on $|P|+|Q|+|R|$:

(i) If $P\parallel R=Q\parallel R$ then $P=Q$.

(ii) If $R\xrightarrow{a}R'$ and $P\parallel R=Q\parallel R'$ then $Q\xrightarrow{a}Q'$ for some $Q'=P$.

(i): Let $P\parallel R=Q\parallel R$. Suppose $P\xrightarrow{a}P'$. Then $P\parallel R\xrightarrow{a}P'\parallel R$; so, there exists $S=P'\parallel R$ such that $Q\parallel R\xrightarrow{a}S$. Hence, either

(a) $\exists Q'$ such that $Q\xrightarrow{a}Q'$ and $Q'\parallel R=P'\parallel R$, or

(b) $\exists R'$ such that $R\xrightarrow{a}R'$ and $Q\parallel R'=P'\parallel R$.

For (a), by induction hypothesis (i), $Q'=P'$. For (b), by induction hypothesis (ii), there exists $Q'=P'$ such that $Q\xrightarrow{a}Q'$. Hence, in any case, there exists $Q'=P'$ such that $Q\xrightarrow{a}Q'$. Similarly, if $Q\xrightarrow{a}Q'$ then there exists $P'=Q'$ such that $P\xrightarrow{a}P'$. Therefore, $P=Q$.

(ii): Let $R\xrightarrow{a}R'$ and $P\parallel R=Q\parallel R'$. Then $P\parallel R\xrightarrow{a}P\parallel R'$; so, there exists $S=P\parallel R'$ such that $Q\parallel R'\xrightarrow{a}S$. Hence, either

(a) $\exists Q'$ such that $Q\xrightarrow{a}Q'$ and $Q'\parallel R'=P\parallel R'$, or

(b) $\exists R''$ such that $R'\xrightarrow{a}R''$ and $Q\parallel R''=P\parallel R'$.

For (a), by induction hypothesis (i), $Q' = P$. For (b), by induction hypothesis (ii), there exists $Q' = P$ such that $Q \xrightarrow{a} Q'$.

Hence, in any case, there exists $Q' = P$ such that $Q \xrightarrow{a} Q'$. □

The main result now follows quite simply. To state it in the simplest form, we understand that 0 is the empty parallel composition of no processes.

**Theorem 2.4** (Unique decomposition of processes). *Any term $P \in \mathscr{P}$ can be expressed uniquely, up to $\sim$, as a parallel composition of primes.*

**Proof.** First, it is easy to see that a prime decomposition exists, not necessarily uniquely, since factorization into non-0 factors decreases the depth; hence, repeated factorization must terminate. For uniqueness, we argue by induction on $|P|$.

Suppose first that $P = Q$, and that $P$ and $Q$ have prime factorizations given by

$$P = C \parallel A_1 \parallel A_2 \parallel \cdots \parallel A_k,$$

$$Q = C \parallel B_1 \parallel B_2 \parallel \cdots \parallel B_l.$$

That is, the two factorizations have a common prime factor. Then by the cancellation lemma (Lemma 2.3), we have

$$A_1 \parallel A_2 \parallel \cdots \parallel A_k = B_1 \parallel B_2 \parallel \cdots \parallel B_l.$$

By the inductive hypothesis, $A_1 \parallel \cdots \parallel A_k$ and $B_1 \parallel \cdots \parallel B_l$ must be identical prime factor decompositions. Thus, the prime factor decompositions for $P$ and $Q$ above are identical.

Now suppose that $P = A_1 \parallel \cdots \parallel A_k$ and $Q = B_1 \parallel \cdots \parallel B_l$ are prime factor decompositions such that for all $i$ and $j$, $A_i \neq B_j$. If $k = 1$ or $l = 1$ then $P = Q$ is prime; so, $k = l = 1$ and $A_1 = B_1$, contradicting the distinctness of the $A_i$ and $B_j$. Hence, assume that $k, l \geqslant 2$, and (w.l.o.g.) that, for all $i$ and $j$, $|A_1| \leqslant |A_i|, |B_j|$. Let $a, R$ be such that $A_1 \xrightarrow{a} R$ and, since $|R| < |A_1| \leqslant |P|$, let $R$'s unique decomposition be

$$R = R_1 \parallel R_2 \parallel \cdots \parallel R_r.$$

Then $P \xrightarrow{a} P'$, with unique decomposition (since $|P'| < |P|$)

$$P' = R_1 \parallel R_2 \parallel \cdots \parallel R_r \parallel A_2 \parallel \cdots \parallel A_k.$$

Now $Q \xrightarrow{a} Q' = P'$; so, for some $B_j$, w.l.o.g. $B_1$, we have $B_1 \xrightarrow{a} T$ and

$$Q' = T \parallel B_2 \parallel \cdots \parallel B_l.$$

Now the decomposition of $P' = Q'$ is unique, and $l \geqslant 2$; so, $B_2$ must be equal to one of $R_1, \ldots, R_r, A_2, \ldots, A_k$. But $B_2 \neq R_p$, $1 \leqslant p \leqslant r$, since $|R_p| < |A_1| \leqslant |B_2|$; so, $B_2$ must be equal to some $A_i$, which contradicts our assumption. □

We now turn to other congruences. A well-known congruence is the *testing equivalence* of de Nicola and Hennessy [2] or, equivalently, the *failures equivalence* of Brookes et al. [1]. We use the failures terminology, as follows:

(1) A set $R \subseteq Act$ is a *refusal set* of $P$ if $P \overset{a}{\not\rightarrow}$ for all $a \in R$.

(2) If $s \in Act^*$, any pair $(s, R) \in Act^* \times 2^{Act}$ is a *failure* of $P$ if, for some $P'$, $P \overset{s}{\rightarrow} P'$ and $R$ is a refusal set of $P'$.

(3) Two processes are *failures-equivalent*, written $=_f$, if they possess the same failures. (This is easily shown to be a congruence.)

As an example, consider $P_1 \equiv a.b.0 + a.c.0$ and $P_2 \equiv a.(b.0 + c.0)$; the pair $(a, \{b\})$ is a failure of $P_1$ but not of $P_2$. This shows that failures equivalence is stronger than traces equivalence (which we consider below). Further, consider $Q_1 \equiv a.b.c.0 + a.b.d.0$ and $Q_2 \equiv a.(b.c.0 + b.d.0)$; it can be seen that $Q_1$ and $Q_2$ have exactly the same failures even though they are not bisimilar; so, bisimilarity is stronger than failures equivalence.

Oddly enough, unique decomposition fails for finite processes under $=_f$. Rob van Glabbeek showed that there are $P_1, P_2, Q_1, Q_2$, all prime for $=_f$, with $P_i \neq_f Q_j$ $(i, j \in \{1, 2\})$, such that

$$P_1 \parallel P_2 =_f Q_1 \parallel Q_2$$

Writing $a.a.a.0$ as $a^3$, etc., he took

$$P_1 \equiv a + a^2, \qquad P_2 \equiv a + a^2,$$

$$Q_1 \equiv a, \qquad Q_2 \equiv a + a^2 + a^3.$$

It is easy to see that $P_i \neq_f Q_j$ $(i, j \in \{1, 2\})$; in fact, they are not even trace-equivalent. By an exhaustive argument they can be proved to be prime. But if we take $\mathrm{Act} = \{a\}$ then $P_1 \parallel P_2$ and $Q_1 \parallel Q_2$ have exactly the same failures as the process $a^2 + a^3 + a^4$, namely

$$(\varepsilon, \emptyset), (a, \emptyset), (a^2, \emptyset), (a^2, \{a\}), (a^3, \emptyset), (a^3, \{a\}), (a^4, \emptyset), (a^4, \{a\}).$$

Now let us consider trace equivalence.

(1) A string $s \in Act^*$ is a *trace* of $P$ if $P \overset{s}{\rightarrow}$.

(2) Two processes are *trace-equivalent*, written $=_t$, if they possess the same traces. (This is also a congruence.)

Each congruence class of processes may be thought of as a finite nonempty prefix-closed set of strings, and under this interpretation $\parallel$ is just the familiar shuffle operator. Note that van Glabbeek's example tells us nothing in this case, because none of $P_1, P_2, Q_2$ is prime for $=_t$; for example $a + a^2 =_t a^2 =_t a \parallel a$.

However, we recently received Joram Hirshfeld's interesting paper [3], in which he studies a rather different notion of decomposition. We cannot see how to relate his results to ours, but a remark in his letter needed only a minor adjustment to show that

unique decomposition *fails* for finite prefixed-closed languages under shuffle. We thank him for the following counter example: $A=\{\varepsilon, a, b\}$ and $B=\{\varepsilon, a, aa, b, bb\}$ are both prime, and $A \| A \| A = A \| B$.

## 3. Infinite processes

Infinite processes can be represented as solutions of equation sets $\{X_i = E_i;\ i \in I\}$, where each $E_i$ is a $\Sigma$-term over the variables $\{X_i;\ i \in I\}$. When $I$ is finite, we have the *finite-state* processes.

Let us write $a^*$ for the process defined by $X = a.X$. Now it is easy to show that decomposition into a *finite* set of prime factors does not exist, in general, for finite-state processes. In fact, if $a^* = P_1 \| \cdots \| P_n$ then one can show that $P_j = a^*$ for some $j$; yet $a^*$ is not prime since, for example, $a^* = a \| a^*$. The question arises: Is $a^*$ in a sense the *only* obstacle to unique decomposition?

From now on we call $Q$ a *derivative* of $P$ if $P \xrightarrow{s} Q$ for some $s \in Act$. Also we write $P \xrightarrow{a} P'$ when, for some $P''$, $P \xrightarrow{a} P'' \sim P'$.

**Definition 3.1.** A process $P$ is *a-impure* if $Q \xrightarrow{a} Q$ for every derivative $Q$ of $P$. $P$ is *impure* if it is $a$-impure for some $a \in Act$; otherwise it is *pure*.

Intuitively, $P$ is $a$-impure if its transition graph – reduced w.r.t. $\sim$ – has a tight loop labelled $a$ at every node. Clearly, if $P'$ is the result of removing all these tight loops then $P = P' \| a^*$ and $P'$ is no longer $a$-impure. So, impurities can be factored out. We conjectured for a while that impurities were indeed the only obstacle to unique decomposition, having failed to find any counterexample; in other words, we thought that pure finite-state processes could be uniquely decomposed, but could find no proof. But Jan Friso Groote has recently shown that this is false; we thank him for the following counterexample. Let $Q \equiv a^* \| b^*$ and $P \equiv a.Q$; then $P$ is pure, but $P = P \| P$.

Groote's example is "nearly" impure; after one action $P$ degenerates into an impure process. It is amusing to note that, for infinite-state processes, the conjecture fails even for processes which never degenerate into impurity. To see this, define $C_0, C_1, \ldots$ as follows:

$$C_0 = up.C_1,$$

$$C_{i+1} = up.C_{i+2} + down.C_i$$

($C_0$ is a simple counter.) Then every $C_i$ is pure. But $C_i = up.down \| C_i$ for each $i$! So, at least for infinite-state processes, there is a wider class of "impurities" to be factored out before we can hope for unique decomposition.

Let us now look at a subclass of pure processes.

**Definition 3.2.** A process $P$ is *a-live* if no derivative $Q$ of $P$ has an infinite transition sequence $Q(\stackrel{a}{\rightarrow})^{\omega}$. $P$ is *live* if it is $a$-live for all $a \in Act$. The *a-life* of $P$ is the largest $k$ for which $Q(\stackrel{a}{\rightarrow})^{k}$ for some derivative of $P$. The *life* of $P$ is the sum of its $a$-lives for $a \in Act$.

Note that, if $P$ is finite-state and $a$-live, then its $a$-life is finite. We are concerned with liveness only for finite-state processes.

Liveness is more tractable than purity, because when $P$ is live then so are its factors, and nontrivial factorization decreases life. We have been able to adapt the proof of our previous cancellation lemma to prove Lemma 3.3

**Lemma 3.3** (Cancellation). *For live finite-state* $P$, $Q$ *and* $R \in \mathscr{P}$,

$$P \| R = Q \| R \text{ implies } P = Q.$$

**Proof** (*Outline*). Corresponding to (i) in the previous cancellation lemma, the main result is achieved by showing that the binary relation

$$\{(P, Q);\ P \| R = Q \| R, \text{ for some } R\}$$

is a *bisimulation* [6, 4] over live finite-state processes. Subsidiary to this proof, and corresponding to (ii) in the previous lemma, we prove that

if $R \stackrel{a}{\rightarrow} R'$ and $P' \| R = Q \| R'$ then $Q \stackrel{a}{\rightarrow} Q'$ for some $Q'$ and $R''$ such that $P' \| R'' = Q' \| R''$.

The proof is by induction on the largest $k$ for which $R(\stackrel{a}{\rightarrow})^{k}$. □

Unfortunately, a similar adaptation does not seem to work for our proof of the unique decomposition theorem; thus, we leave the unique decomposition of live finite-state processes as a conjecture.

## References

[1] S.D. Brookes, C.A.R. Hoare and A.W. Roscoe, A theory of communicating sequential processes, *J. ACM* **31** (1984) 560–599.

[2] R. de Nicola and M.C. Hennessy, Testing equivalence for processes, *Theoret. Comput. Sci.* **34** (1983) 83–133.

[3] J. Hirshfeld, Deterministic concurrent systems, Tech. Report 162/90, Institute of Computer Sciences, Tel Aviv University, 1990.

[4] R. Milner, *Communication and Concurrency* (Prentice Hall, Englewood Cliffs, NJ, 1989).

[5] F. Moller, Axioms for concurrency, Ph.D. Thesis, Report ECS-LFCS-89-84, Computer Science Department, University of Edinburgh, 1989.

[6] D.M.R. Park, Concurrency and automata on infinite sequences, in: Lecture Notes in Computer Science, Vol. 104 (Springer, Berlin, 1980) 167–183.